UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/675,116	09/28/2000	Grzegorz J. Czajkowski	SUN-P5075-RSH	9136

22835 7590 01/19/2005

c/o A. RICHARD PARK, REG. NO. 41241
PARK, VAUGHAN & FLEMING LLP
2820 FIFTH STREET
DAVIS, CA 95616

EXAMINER	
ALI, SYED J	

ART UNIT	PAPER NUMBER
2127	

DATE MAILED: 01/19/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/675,116	CZAJKOWSKI ET AL.	
	Examiner	Art Unit	
	Syed J Ali	2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 June 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4,6-11,13-18 and 20-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4,6-11,13-18 and 20-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to the amendment filed June 4, 2004. Claims 1-4, 6-11, 13-18, and 20-24 are presented for examination.

2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Rejections - 35 USC § 103

3. **Claims 1-4, 6-11, 13-18, and 20-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gosling (USPN 5,668,999) in view of Jagannathan et al. (USPN 6,496,871) (hereinafter Jagannathan).**

4. As per claim 1, Gosling teaches the invention as claimed, including a method for verifying type safety of an application snapshot, the method comprising:

the application snapshot includes a subprogram, an operand stack, and a point of execution (col. 6 lines 28-46);

examining the application snapshot on the second computing device to identify the subprogram and the point of execution within the subprogram (col. 7 lines 31-44);

examining the subprogram on the second computing device to determine an expected structure of the operand stack at the point of execution (col. 7 line 10-65);

validating that the state of the application snapshot on the second computing device is consistent with the expected structure of the operand stack (col. 7 line 58-65);

verifying on the second computing device that variables and arguments within the application snapshot are of the proper type (col. 10 lines 16-55); and

if the state of the application snapshot is validated as consistent with the expected structure of the operand stack, executing the application snapshot on the second computing device (col. 10 lines 56-64).

5. Jagannathan teaches the invention as claimed, including the following limitations not shown by Gosling:

the application snapshot including a state of an executing program that is moved from a first computing device to a second computing device across a network in order to continue execution on the second computing device (col. 20 lines 45-55);

receiving the application snapshot of the executing program from the first computing device on the second computing device (col. 18 lines 4-23);

resuming execution of the application snapshot at the point of execution on the first computing device (col. 18 lines 4-23).

6. It would have been obvious to one of ordinary skill in the art to combine Gosling and Jagannathan since Gosling, while providing means for verifying an application before execution, fails to specify how migration of an application might be handled. Rather, the procedure for verifying an application snapshot is presented, but an assumption is made that once the program begins executing, it will continue to reside on that machine. With the advent of mobile code and distributed processing, a method of handling process migration during process execution is necessary. Additionally, since many networks are heterogeneous, verification across platforms is necessary. Jagannathan provides a system that allows dynamic process migration while also

maintaining state information related to the ongoing execution. Jagannathan acknowledges that the problem of migration of ongoing processes has been addressed, but the prior art does not allow for state information related to those processes to be easily migrated (col. 4 lines 47-59). Jagannathan also indicates that process migration may occur between heterogeneous machines, indicating a specific need for a verification method, such as the one disclosed by Gosling (col. 18 lines 4-23). Jagannathan seeks to improve the prior art by not only providing a system that allows migration of ongoing processes, but also allows the state information to be migrated (col. 5 lines 28-35). The combination of Gosling and Jagannathan would provide an exemplary model for verifying the type safety of an executing application, while maintaining state information related to the application as it is migrated from one machine to another.

7. As per claim 2, Gosling teaches the invention as claimed, including the method of claim 1, wherein examining the subprogram to determine the expected structure of the operand stack at the point of execution involves examining the subprogram with a code verifier, wherein the code verifier ensures that:

the subprogram does not cause the operand stack to overflow and underflow (col. 8 line 46 - col. 9 line 18);

a use of a local variable does not violate type safety (col. 10 lines 16-27); and

an argument of an instruction is of an expected type (col. 6 lines 6-13).

8. As per claim 3, Gosling teaches the invention as claimed, including the method of claim 1, wherein the operand stack contains at least one local variable, at least one argument that is

passed as a parameter to the subprogram, and an offset to the point of execution within the subprogram (col. 5 lines 21-29; col. 6 lines 6-13; col. 6 lines 28-46).

9. As per claim 4, Gosling teaches the invention as claimed, including the method of claim 2, wherein the expected structure of the operand stack includes a collective size of entries and the types of entries expected on the operand stack at the point of execution within the subprogram (col. 7 lines 20-30).

10. As per claim 6, Gosling teaches the invention as claimed, including the method of claim 4, wherein validating that the state of the application snapshot on the second computing device is consistent with the expected structure of the operand stack involves ensuring that the collective size of entries and the types of entries on the operand stack agree with the collective size of entries and the types of entries expected on the operand stack (col. 9 lines 44-62).

11. As per claim 7, Jagannathan teaches the invention as claimed, including the method of claim 1, wherein resuming execution of the application snapshot involves restarting the subprogram at the point of execution within the second computing device (col. 18 lines 4-23).

12. As per claim 22, Jagannathan teaches the invention as claimed, including the method of claim 1, further comprising restoring the state of an object within the application snapshot on the second computing device by changing a pointer from an address of the object on the first

computing device to an address of the object on the second computing device (col. 21 lines 7-27).

13. As per claims 8-11, 13-18, 20-21, and 23-24, Gosling teaches the invention as claimed, including an apparatus including a computer-readable storage medium storing instructions that when executed by a computer causes the computer to perform the method of claims 1-4, 6-7, and 22 (Fig. 2).

Response to Arguments

14. Applicant's arguments filed June 4, 2004 have been fully considered but they are not persuasive.

15. Applicant argues on page 9, "*Gosling is clearly directed to a system 'for verifying the proper operation of the executable program prior to actual execution by a host processor' [whereas] ...the present invention is directed to forming a snapshot of an executing program.*" Applicant adds that "*verifying a bytecode program...is not the same as verifying a snapshot of an executing program.*"

Applicant further argues that "*Gosling verifies only the bytecode program, and not the entire state of the executing program. Hence, Gosling does not validate variables and arguments.*" Applicant also argues that Jagannathan fails to resolve the alleged deficiencies of Gosling.

16. Applicant's argument relating to Gosling's alleged inability to verify an executing program is deficient on the grounds that Gosling's disclosure that the bytecode program is verified "prior to actual execution" is performed in exactly the same manner as the claimed snapshot verification. Gosling downloads a bytecode program and verifies the type safety of the program including a snapshot of the application at various points of execution before executing it. If the preprocessing verification generates any program faults, the program is not executed.

Turning to claim 1 of the present invention, a snapshot is taken of an executing program and moved to a second computing device. Thereafter, the type safety is validated along with the expected structure of the operating stack. After all the verifications have been performed and certified, the execution is resumed, i.e. execution begins. The execution of the program being verified is temporarily suspended while it is being verified, and execution resumes only if the state of the application is verified as being consistent with the expected structure of the operand stack. This manner of verifying a snapshot *before resuming execution* functions in the same way that Gosling pre-verifies stack usage of a bytecode program *before commencing execution* on a processor.

Furthermore, Gosling specifically addresses the verification of ongoing programs to verify that all variables and arguments of a program are consistent with what is expected to be on the operand stack at the particular execution point (col. 2 lines 32-39, "**During processing of the specified program**, the verifier takes a 'snapshot' of the virtual operand stack immediately prior to each multiple-entry point..., compares that snapshot with the virtual operand stack state after processing each of the other preceding bytecode instructions for the same multiple-entry point, and generates a program fault if the virtual stack states are not identical") (emphasis added).

While Gosling generally verifies bytecode programs before any execution has started, methods are provided for verifying programs that may be executing. Gosling is primarily concerned with pre-verification of bytecode programs, and does not specifically address such issues as process mobility or migration of processes. Jagannathan is cited as showing why distributed processing systems may move an executing program from one machine to another (Abstract, "objects are selectively movable among the at least two computer machines by a programmer of the system"), while Gosling addresses the need for providing security of such mobile programs (col. 5 lines 5-12, "the downloaded bytecode program may contain errors involving the data types of operands not matching operands, which may cause the program to fail during execution. Even worse, a bytecode program might attempt to create object references...and to thereby breach the security and/or integrity of the user's computer").

Conclusion

17. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (571) 272-3769. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Syed Ali
December 17, 2004

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100